INTEGRATION OF STOCKYARD AND RAIL NETWORK: A SCHEDULING CASE STUDY

Amir Abdekhodaee, Simon Dunstall, Andreas T. Ernst and Lam Lam

CSIRO Mathematical and Information Sciences
P.O.Box 10, Clayton South, Vic 3169, Australia

{Amir.Abdekhodaee, Simon.Dunstall, Andreas.Ernst, Lam.Lam}@csiro.au

ABSTRACT

We investigate the integration of scheduling for a rail system and some of the operations in a coal terminal system, based on representative data. The rail and terminal systems are tightly-coupled and experience a high service demand from the expensive infrastructure. This makes efficient operations essential. However no central authority is in a position to effectively create global schedules across the whole system making a collaborative scheduling approach necessary. We provide mixed integer programming models for the two systems and discuss the merits and disadvantages of devising such an approach. The emphasis is on developing methods that promote coordination between the operational scheduling functions of the two systems.

Key Words: Supply Chain Management, Scheduling, Combinatorial Optimisation.

1. INTRODUCTION

Many large mining operations in Australia and other parts of the world rely on multi-stage bulk-material transport systems to transport material from mines to terminals, for shipment or further processing. Operating these supply chains efficiently requires a series of scheduling problems to be solved. If railways and ports are part of the bulk-material transport system then foremost amongst these scheduling problems are train timetabling, the assignment of rail vehicles (locomotives and rollingstock) to trains, stockyard management, ship handling (berthing) and loading operations, and the rostering of crews to provide such services.

Our focus in this paper is on the integration of operational (or demand responsive) train timetabling and stockyard management in a port terminal. We present a description of relevant railway systems and stockyards, present mixed integer programming models for each of these components of a coal supply chain, and investigate solution methods for the operational scheduling problem. In doing so we report on progress towards a whole-of-supply-network solution approach where our fundamental strategy is to treat each sub-problem and its solution method as a module. In such an approach the operation of the modules is orchestrated for the purpose of generating feasible and near-optimal solutions to the problem.

The idea of a ‘modular’ or ‘decomposition’ approaches will not be unfamiliar to the industrial OR practitioner. The attractiveness of such approaches can be attributable to various factors. For practicality or expediency, we may wish to exploit known algorithms and existing software that solves certain sub-problems. Furthermore, we may find that ‘combining’ existing models for sub-
systems (so as to form formulations for a larger problem) may not be possible. For example, a certain formulation for a locomotive rostering problem might require problem instances to be derived from a known train timetable. This will preclude the simultaneous generation of timetables and rosters if we choose to make use of this formulation. When addressing large and complex problems it is also often the case that the problem as a whole is too difficult to solve by means other than a modular approach.

The rail system to be considered in this paper consists of a tree-structured network of railway track that connects a number of mines to a single terminal. We seek efficient schedules for the railing of materials from these mines to the terminal. In our chosen abstraction of the problem the planning period (i.e. the period for which a schedule is to be produced) is in the order of a week. Prior to the start of the planning period, the operator of the rail system receives requests for train services. A request states a customer’s wish for a train to load material at a specific mine and deliver it to the terminal. The request will also state a preferred delivery time, i.e. an arrival time at the terminal. We refer to the set of requests as the demand. The terminal has two major responsibilities: to receive various types of coal in various quantities, via the railing system, and to deliver these products to incoming ships. The terminal consists mainly of open storage areas called stockpiles. We are concerned in this paper with the train timetabling and stockyard management problems, as part of a wider exploration into scheduling processes in this supply chain.

Most of the existing literature on scheduling comparable rail operations focuses on particular aspects of the overall problem. For a relatively recent survey of this literature, the reader may consult Ferreira (1997). The type of rail network considered in this paper and commonly found in the context of the Australian mining industry, is characterised by a tree structure and large sections of single-line track with occasional passing loops. This means that train-routing is not an issue. Instead, train scheduling solutions need to focus on how to best work within the capacity limitations imposed by the single track sections.

Previous work in this area includes determining the optimal location of sidings in the developing or expanding a single line railway track Higgins et al. (1997), and methods for scheduling given a single track rail system (Cai et al. (1998), Higgins (1997)). The locomotive assignment problem has been studied for some time with early work by Bartlett in the 1950s (Bartlett (1957), Bartlett and Charnes (1957)) through to more recent papers considering different variants of the basic problem depending on the number of locomotive classes and other side constraints Al-Amin et al. (1999), Cordeau et al. (2000), Noble et al. (2001), Ziarati et al. (1999). Some effort has also been made in applying these methods in practice to improve railway operations (Ahuja et al. (2002), Nou et al. (1997), Ziarati et al. (1997)). More recently, He et al. (2003) provided a dispatcher model for rail yard activities involved with classification and assembly of trains. They provide a mixed integer programming and suggest ways to overcome the complexity of the general complexity of the problem. Ballis and Golias (2004) present a modelling approach based on an integrated framework for an expert system and a detailed simulation model to compare conventional and advanced railroad terminal equipment.

2. OVERALL COAL SUPPLY CHAIN

The overall supply chain for coal export includes the following main components: users (or mining companies), mines, railing system, terminal system, overseas purchasers and offshore transportation
Figure 1. A schematic representation of a coal export supply chain

system (ships). There are a number of coal mining companies that manage the mining, production and exporting of coal from different mines. These mining companies share some export facilities, including the railing system and coal export terminal (Figure 1).

A typical user has several mines and can provide a customer with a variety of coal types. They negotiate on the price, amount and type of product and particularly over the time window(s) that product should be available for shipping from the terminal. The time-window information is communicated to mines, the railing system and the coal terminal by the users, and to the offshore transportation system by the overseas purchaser. All the coal of different types and from all the users are inloaded, collected, stacked, reclaimed and outloaded onto the ships by the coal terminal. The coal terminal is owned by users and the government, and operated for (and on behalf of) the various stakeholders.

There are different types of coal in different quantities coming from various mines by trains to the coal terminal. The coal is unloaded via two inloaders and then is transferred to a stockyard via a network of conveyors. Subsequently it is transported to ships waiting to be loaded. The terminal can accommodate the loading of two ships at a time. A ship might carry more than one particular product, and ship loading will commence only when the required coal types and quantities are ready at the terminal. In a sense the coal terminal acts as a large multi-product buffer between incoming trains and outgoing ships.

3. RAILING MODEL

Throughout this paper, we assume that the railway infrastructure consists of a single unloading terminal, a single rollingstock depot, one or more crew depots and multiple mines. All trains originate and terminate at the rollingstock depot, and visit one mine and the terminal en-route (Figure 2).
For the operational scheduling problem we have two categories of input: information regarding the demand and information about the physical system over which the coal is transported. For the physical system we have many parameters that specify the capacity and connectivity of the system of railway track and the capabilities and availability of rollingstock and train crew.

The demand is represented by a set (array) of \( N \) individual demands for train services. The \( k \)th train service is represented by the tuple \( (m_k, \theta_k, T_k) \). This \( k \)th service transports a mass \( m_k \) of a product specified by \( \theta_k \). The product specification \( \theta_k \) determines the mine (coal loading point) and the type of coal to be transported. The value of \( T_k \) is the desired arrival-time of the \( k \)th train service at the unloading terminal. An individual demand is a service request asked of the rail system by the terminal operator, on behalf of a mining company, in our abstraction of the reality of the coal transportation business.

We define a train as an assemblage of rollingstock undertaking a discrete journey and fulfilling a specific demand \( (m_k, \theta_k, T_k) \). In our model a journey is composed of two trips. The first trip is from the rollingstock depot to a mine and the second trip is from the mine back to the rollingstock depot via the unloading terminal. Items of rollingstock may be marshalled together and undertake several journeys before being re-marshalled; each such journey gives rise to a distinct and unique train.

For modelling purposes we divide the network of railway tracks into sections, or segments. This style of representation is the basis of typical safe-working regulations for railway operation. In general, safe-working regulations allow at most one train per section at any time. We seek to create schedules that prescribe train movements that are legal with respect to such regulations. However, due to the manner in which we wish to describe train progress in our model, the capacity of a segment will exceed one in parts of the network that have certain characteristics.

The position of a train in the network is specified by the segment it occupies and the direction it is travelling in. In accordance with custom, we state the direction of travel as being either up or down. It is not important from a mathematical viewpoint which direction on a particular segment is considered as ‘up’ and which is considered as ‘down’. We describe the rail network mathematically...
as a directed graph. In this network graph, each node corresponds uniquely to a position, and the arcs join positions. If a train can travel from position $i$ to position $j$ without having to pass through another (intermediate) segment, a directed arc joins the node for position $i$ with that for position $j$. We describe the (spatial) passage of a train through the network as a sequence of positions, and for feasibility this sequence must correspond to a valid path in the network graph. This sequence is the route of a train trip.

We assume that the route for each trip is determined prior to the start of the scheduling process, and so is part of the input data. Specifically, in our model a route for a train can be uniquely determined from knowledge of $\theta_k$. This follows from two assumptions. First, we assume that all trains originate and terminate at the rollingstock depot and visit one mine and the terminal, and second, we assume that there is only one permissible path between each of these physical locations. We assert that this second assumption is reasonable for the tree-like networks of interest in this paper. In our implementation of the model the input data contains a route-map which specifies the route between the rollingstock depot and each mine, and the route from each mine back to the rollingstock depot via the unloading terminal.

The single-permissible-path assumption also gives rise to the aforementioned difference between the typical safe-working definition of a segment and the definition adopted for our scheduling model. From Figure 2, it can be observed that the ‘terminal’ segment consists of two parallel roads. We allow both roads to be occupied simultaneously, and so this segment is assigned a capacity of two trains. This allows us to define a route without specifying the particular portion of track used in a segment (such as the ‘terminal’ segment). The resulting schedule will be feasible according to safe-working regulations, as long as the track layout in the segment passes an application of common-sense. In the case of the ‘terminal’ segment, the presence of single entry and exit tracks (each with a capacity of one train) is sufficient for safe-working feasibility.

Note that one train could be scheduled to overtake another at the ‘terminal’ segment. In general, our model allows overtaking of trains, through appropriate modelling of segments. However, we refrain from modelling overtaking points in this way, except at the terminal or mines with a loading capacity of more than one train. This is because the presence of such segments increases the difficulty of solving the timetabling problem.

In our model, a train occupies one segment at any instant. Obviously, this is an approximation that can lead to deviations between feasibility in the model and feasibility in practice. Trains are of a considerable length and will occupy two segments simultaneously when they progress from one segment to another. However, in the coal-transporting supply-chains of immediate interest, the length of a segment will typically be sufficient to render our approximation reasonable.

The time taken for a train to traverse a segment we refer to as the transit time. Transit times are part of the input to the operational scheduling problem, and in general we allow this time to be train- and direction-specific. This allows us to differentiate between loaded and unloaded trains of different length, weight and haulage power. We may also impose a minimum inter-arrival time on a segment: this restricts the entry time (from any direction) of a train into a segment to be no less than a certain amount of time after the exit time of a preceding train that occupies that segment.

Inter-arrival times can be used to compensate for train-length in segment-to-segment progression. In the network models that we have used in computational experiments, we have applied non-zero inter-arrival times only at mines. This particular use of inter-arrival times is motivated by
in-practice constraints arising from matters of operational policy and of equipment-use limitations.

4. TIMETABLING FORMULATION

The timetabling solver implements a MIP formulation of the timetabling problem. In this formulation we denote the set of track segments by $B$ and the set of roundtrips by $R$. The path (or route) for a roundtrip $R_r \in R$ is given by $S_r$. $S_r$ is a sequence of positions that starts from the first segment out of the depot (position $S_{r,1}$) and ends at the depot (position $S_{r,|S_r|}$). Thus $|S_r| > 1$ and $S_{i,|S_i|} = S_{j,|S_j|}$ for all pairs of roundtrips $(i, j)$. A position is a segment–direction pair: we will denote a position by a tuple $(\beta, \gamma)$ where $\beta \in B$ is the segment and $\gamma$ is the direction. In the formulation, the variable $t_{r,s}$ represents the time that $R_r \in R$ enters position $s \in S_r$. The variable $d_r$ is the total duration of the roundtrip $R_r \in R$, and the value of $d_r$ is given by $t_{r,|S_r|} - t_{r,1}$ for all $R_r \in R$.

We define $\pi_{r,s}$ as the position $(\beta_{r,s}, \gamma_{r,s})$ of the $s$th step of roundtrip $R_r \in R$, and $\phi_r(\beta, \gamma)$ as a function returning the index of position $(\beta, \gamma)$ in the path for roundtrip $r$. In general, a position may appear more than once in a roundtrip’s path, and one can formulate the problem so that it handles this general case. For simplicity, however, we will assume that a position appears at most once in a roundtrip’s path.

We define $x_{i,u;j,v}$ as a binary variable indicating whether train $i$ starts its $u$th step ($S_{i,u}$) no later than when train $j$ starts its $v$th step ($S_{j,v}$). That is, $x_{i,u;j,v}$ equals one if $t_{i,u} \leq t_{j,v}$ and zero otherwise. Variable $x_{i,u;j,v}$ will only be defined if $i < j$ and $\beta_{i,u} = \beta_{j,v}$. The restriction $i < j$ allows us to minimise the number of $x$ variables in the formulation.

We define $\beta(x)$ as the segment addressed by a variable $x$. For example, $\beta(x_{i,u;j,v})$ equals segment $\beta_{i,u}$. We also denote by $B_1$ the maximal subset of segments with a capacity of one train ($B_1 \subseteq B$).

The set of all $x$ variables is $\mathcal{X}$. We define $\mathcal{X}_1$ as the subset of $\mathcal{X}$ variables for which segment $\beta(x \in \mathcal{X}_1)$ has capacity one (i.e. $\beta(x) \in B_1$ if $x \in \mathcal{X}_1$). We define $\mathcal{X}_D$ as the set of $\mathcal{X}$ variables for which $\beta(x \in \mathcal{X}_D)$ is the depot segment (with infinite capacity), and define $\mathcal{X}_M$ such that $\beta(x \in \mathcal{X}_M)$ has a finite capacity that is greater than one, so that $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_D \cup \mathcal{X}_M$.

The train timetabling objective is comprised of two components, each of which measure deviation from an ‘ideal’ timetable. These components are the timing-point deviation and the journey-duration deviation.

Each roundtrip $R_i \in R$ has an target duration $d_t^i$. A penalty is recorded in the objective function if the duration of a journey differs from this target duration. The penalty is calculated relative to a piecewise-continuous and convex function $f^D_i$ that is specific to the journey (and thus is part of the input data). The function $f^D_i$ is given by the sum of an arbitrary number of ‘earliness’ and ‘tardiness’ functions. An earliness function, for journey duration, takes the form

$$\alpha \cdot \max \left\{ 0, d_i^* - (d_i + c) \right\}$$

where $d_i$ is the journey time and $c$ is an ‘offset’. The offset has the effect of permitting the actual journey duration to be less than the target journey time by an amount $c$ before a penalty is recorded. An objective function penalty of $\alpha$ per time unit is exacted if the journey takes less than $d_i^* - c$ time units to complete. The penalty rate $\alpha$ may differ between one earliness function and the next. Tardiness functions have a symmetric form, i.e. $\alpha \cdot \max \left\{ 0, d_i - (d_i^* + c) \right\}$.

In addition to a target journey duration, each roundtrip has one or more timing points. The set of timing points for a roundtrip $R_i$ is denoted by $\Omega_i$. Each timing point of $R_i$ is associated with a step
in its path $S_i$ and a target time at which the train should begin this step (i.e. enter the corresponding section). No step in the path for a roundtrip can be associated with more than one timing point. Thus any given timing point that is a member of $\Omega_i$ can be unambiguously denoted by $\Omega_{i,u}$, where the timing point in question is associated with step $S_{i,u}$. A set of timing points defines a ‘preferred’ or ‘ideal’ schedule (or partial schedule) for a roundtrip. Every roundtrip has at least one timing point, and one of the timing points for every roundtrip will correspond to the arrival time at the terminal.

An earliness-lateness penalty, of the same form as that for journey duration, is applied for each timing point: a train should enter the segment from the given direction at the target time in order to avoid a penalty. We denote the penalty function for $\Omega_{i,u}$ by $f^T_{i,u}$. The set of timing points and penalty functions is specified in the input data.

In the MIP, the penalty for each earliness or lateness function is represented by a continuous variable, say $\rho$. Variable $\rho$ is constrained as follows for an earliness function on journey duration:

$$\rho \geq \alpha (d^*_k - (d_k + c))$$

$$\rho \geq 0$$

The objective function seeks to minimise the linear sum of these penalties. Thus these two inequalities are sufficient when formulating the MIP. The MIP formulation of the timetabling model is as follows.

**Objective Function**

$$\min \sum_{R_i \in \mathcal{R}} \left( f^D_i(d_i) + \sum_{\Omega_{i,u} \in \Omega_i} f^T_{i,u}(t_{i,u}) \right)$$

**Trip Duration Constraints**

$$d_i = t_{i,|S_i|} - t_{i,1} \quad \forall R_i \in \mathcal{R}$$

**Progression Constraints**

$$t_{i,u+1} \leq t_{i,u} + \tau_{i,u} \quad \forall R_i \in \mathcal{R}, \forall u < |S_f|$$

where $\tau_{i,u}$ is the time taken by $i$ to traverse segment $\beta_{i,u}$.

**Depot Arrivals**

$$t_{j,|S_i|} \geq t_{i,|S_i|} + I_{i,|S_i|} - T_\infty(1 - x_{i,|S_i|,j,|S_j|}) \quad \forall x_{i,|S_i|,j,|S_j|} \in \mathcal{X}_D$$

$$t_{i,|S_i|} \geq t_{j,|S_i|} + I_{j,|S_i|} - T_\infty(x_{i,|S_i|,j,|S_j|}) \quad \forall x_{i,|S_i|,j,|S_j|} \in \mathcal{X}_D$$

where $I_{i,|S_i|}$ is the inter-arrival time at the depot for roundtrip $R_i$: no train can enter the depot within $I_{i,|S_i|}$ time units of the arrival of $R_i$ at the depot. $T_\infty$ is a suitably large constant.

**Occupancy of Single-Capacity Segments**

$$t_{j,v} \geq t_{i,u+1} + I_{i,u} - T_\infty(1 - x_{i,u,j,v}) \quad \forall x_{i,u,j,v} \in \mathcal{X}_1$$

$$t_{i,u} \geq t_{j,v+1} + I_{j,v} - T_\infty(x_{i,u,j,v}) \quad \forall x_{i,u,j,v} \in \mathcal{X}_1$$

25.5.7
where $I_{r,s}$ is an interarrival time: a train cannot enter $\pi_{r,s}$ within $I_{r,s}$ time units of the exit of train $r$ from $\pi_{r,s}$.

**Occupancy of Multiple-Capacity Segments**

\[
\begin{align*}
t_{j,v} & \geq t_{i,u} - T_{\infty}(1 - x_{i,u,j,v}) \quad \forall x_{i,u,j,v} \in \mathcal{X}_M \\
t_{i,u} & \geq t_{j,v} - T_{\infty}x_{i,u,j,v} \quad \forall x_{i,u,j,v} \in \mathcal{X}_M \\
t_{j,v+1} & \geq t_{i,u+1} - T_{\infty}(1 - y_{i,u,j,v}) \quad \forall y_{i,u,j,v} \in \mathcal{Y} \\
t_{i,u+1} & \geq t_{j,v+1} - T_{\infty}y_{i,u,j,v} \quad \forall y_{i,u,j,v} \in \mathcal{Y} \\
m_{i,u} & \geq \left( \sum_{j,v:x_{i,u,j,v} \in \mathcal{X}_M} x_{i,u,j,v} + \sum_{j,v:y_{i,u,j,v} \in \mathcal{Y}} y_{i,u,j,v} - \sum_{j,v:y_{i,u,j,v} \in \mathcal{Y}} (1 - y_{i,u,j,v}) \right) \quad \forall i, u : m_{i,u} > 1 \quad (1)
\end{align*}
\]

where $m_{i,u}$ is the capacity of segment $\beta_{i,u}$. Variable $y_{i,u,j,v}$ is binary and is defined similarly to $x_{i,u,j,v}$, except that $y_{i,u,j,v}$ equals one if and only if $R_i$ exits $\beta_{i,u}$ no later than when $R_j$ exits $\beta_{i,u}$. There is a one-to-one correspondence between members of $\mathcal{X}_M$ and members of $\mathcal{Y}$.

Constraint 1 ensures that the capacity restriction of a segment is satisfied. This is done by counting all of the trains that enter $\beta_{i,u}$ at or before the time that $R_i$ enters, and all of the trains that exit $\beta_{i,u}$ at or before the time that $R_i$ exits, and ensuring that the difference between these sums is no more than the capacity of the segment. We note that the formulation does not allow for interarrival times on multiple-capacity segments.

**Progression-Inferred Ordering**

\[
\begin{align*}
x_{i,u,j,w} & \geq x_{i,u,j,v} \quad \forall R_i, R_j \in \mathcal{R} : i < j \\
& \quad \forall u, v, w : \beta_{i,u} = \beta_{j,v} = \beta_{j,w}, \\
& \quad v < w, \\
& \quad \beta_{i,u} \in \mathcal{B}_1 \\
x_{i,w,j,v} & \leq x_{i,u,j,v} \quad \forall R_i, R_j \in \mathcal{R} : i < j \\
& \quad \forall u, v, w : \beta_{i,u} = \beta_{i,w} = \beta_{j,v}, \\
& \quad v < w, \\
& \quad \beta_{i,u} \in \mathcal{B}_1
\end{align*}
\]

These constraints are best explained using an example. Let segment $B_b \in \mathcal{B}_1$ be in the paths of an arbitrary pair of roundtrips $R_i$ and $R_j$, where $i < j$. Furthermore, let $S_j$ contain positions ($B_b, up$) and ($B_b, down$). Without loss of generality, assume that $u = \phi_i(B_b, \gamma_i)$ and $v = \phi_j(B_b, \gamma_j) < \phi_j(B_b, \gamma_j) = w$, where $\gamma_i$ and $\gamma_j$ are arbitrary direction values. Clearly, if train $i$ reaches step $u$ before train $j$ reaches step $v$, then train $i$ also reaches step $u$ before train $j$ reaches a later step $w$. In other words, $x_{i,u,j,v} = 1 \Rightarrow x_{i,u,j,w} = 1$. In the absence of additional information, it also follows that nothing can be said about the value of $x_{i,u,j,w}$ when $x_{i,u,j,v} = 0$.

**Transitivity**

\[
\begin{align*}
x_{i,u,k,w} & \geq x_{i,u,j,v} + x_{j,v,k,w} - 1 \quad \forall R_i, R_j, R_k \in \mathcal{R} : i < j < k, \\
& \quad \forall u, v, w : \beta_{i,u} = \beta_{j,v} = \beta_{k,w}, \\
& \quad \beta_{i,u} \in \mathcal{B}_1 \\
x_{i,u,k,w} & \leq x_{i,u,j,v} + x_{j,v,k,w}
\end{align*}
\]
Arbitrarily select three roundtrips $R_i$, $R_j$ and $R_k$ from $\mathcal{R}$. Without loss of generality, assume that $i < j < k$. Let $B_b$ be a segment that is common to all three roundtrips, and let $u$, $v$ and $w$ be indices of $B_b$ in the paths $S_i$, $S_j$ and $S_k$ respectively (there may be two possible values for each of these indices if $B_b$ appears twice in a path). If $t_{i,u} \leq t_{j,v}$ and $t_{j,v} \leq t_{k,w}$, then it must hold that $t_{i,u} \leq t_{k,w}$.

There can be $O(|\mathcal{B}| \cdot |\mathcal{R}|^3)$ of these constraints, thus including them all in a formulation can be prohibitive.

**Same Subroute**

$$x_{i,u-1,j,v-1} = x_{i,u,j,v}$$

If the paths of roundtrips $R_i$ and $R_j$ have two successive positions in common, i.e. $\{S_{i,u-1}, S_{i,u}\} = \{S_{j,v-1}, S_{j,v}\}$, and the capacity of the segment $b = \beta_{i,u-1}$ has a capacity of one, then if roundtrip $R_i$ enters $b$ before roundtrip $R_j$ does, then $R_i$ must also enter the next position before $R_j$ does. This is because overtaking in segment $b$ is not possible. This gives rise to the constraint, which applies to all successive pairs of positions that are common between any pair of roundtrips $R_i$ and $R_j$, as long as the segment of the first position has a capacity of one train.

This constraint can be used to eliminate variables in the formulation, as an alternative to the constraint being explicitly included in the formulation. The variable $x_{i,u-1,j,v-1}$ would be replaced by variable $x_{i,u,j,v}$.

**Opposing Subroute**

$$x_{i,u-1,j,v} = x_{i,u,j,v-1}$$

This constraint is very similar to the ‘Same Subroute’ constraint. An ‘opposing subroute’ is such that $R_i$ visits segment $\beta_{i,u-1}$ in step $u - 1$ and segment $\beta_{i,u}$ in step $v$, whereas $R_j$ visits $\beta_{i,u}$ in step $v - 1$ and $\beta_{i,u-1}$ in step $v$. Thus one train follows a section of the path of the other train, but in the opposite direction. We apply the restriction that at least one of the segments must have capacity one. This implies that if a network has two adjacent segments with a capacity greater than one, then we assume that trains can move between these segments with no chance of conflicting with another train. As with the ‘Same Subroute’ constraint, we can use the relationship of this constraint to substitute variable $x_{i,u,j,v-1}$ for variable $x_{i,u-1,j,v}$.

We have evaluated this formulation within a solution process that iteratively solves the timetabling, locomotive rostering and crew rostering problems in a search for a complete operational schedule for a rail network. The locomotive rostering module was based on an extension of the work of Dunstall et al. (2004), and for crew rostering we made use of a heuristic. Using a single 667 MHz processor of a DEC Alpha computer, we required 45 seconds in total to solve an instance with 16 segments, 10 journeys and 2 mines, and 180 seconds to solve an instance with 91 segments, 20 journeys and 17 mines. The data for the latter test was sourced from information related to approximately a day of operations for an Australian coal supply chain.
From observing these and similar tests we can confidently conclude that the execution time for the solution of the timetabling MIP accounts for at least three quarters of the overall execution time (for the rail operational scheduling problem), and moreover, for the dramatic rise in execution time with instance size. These results indicate that the integer programming approach may not be a viable solution method for a realistic data set.

4.1. A Greedy Heuristic for Railing System

The above integer programming approach works well for small instances. However in order to allow solution of larger instances in a simulation environment (see Section 6.), a faster, more computationally efficient solution method is required. Hence we developed a greedy heuristic that would allow ‘good’ solutions to be constructed quickly.

The basic greedy algorithm is as follows:

1. Step through time in chronological order.
2. Determine which mine/product to dispatch the next available consist to.
3. Simulate (calculate) the time when the train will arrive at the mine, terminal and back at the depot, taking into account delays throughout the system.
4. Return to step 2, dispatching the next consist as soon as it becomes available.

The essential rationale for this approach is that sending off trains as soon as possible and minimising idle time at the depot to maximise throughput. Several variants of this heuristic are possible depending on the method for determining the next train to dispatch. In this paper we compare the following basic methods:

**Railing to Entitlement:** Here the trains are allocated to different mines based on a fixed percentage called the entitlement, provided storage space is available for the terminal for the product of the mine. This is the most ‘fair’ approach from the perspective of the user of the terminal but not necessarily the best from a throughput point of view.

**Railing to Shipping:** This policy only allows trains to be sent for products that need to be loaded onto ships within the next 12 days. The 12-day period is the time span that has been frequently used by operation managers in the terminal that we were involved with. However among all coal products that still need to be railed for this forecast shipping demand, the trains are still allocated based on entitlement.

**Campaign Railing:** This is a purely demand driven method of allocating trains in which the next train is allocated to the first ship in the 12 day look ahead period that does not already have all of its product railed. To prevent extreme congestion effects each mine has a minimum train interarrival time that specifies the minimum time between two consecutive trains being dispatched from the depot. If the last train to a mine has been sent off more recently than this minimum time window, then the mine is skipped and a train sent to the next mine in the list.

The performance of these different heuristic railing methods is compared in Section 6..
5. STOCKYARD SYSTEM

The terminal has two major responsibilities: to receive various types of coal in various quantities via the railing system and to deliver these products to incoming ships. The terminal consists mainly of storage areas called stockpiles which are arranged in rows as depicted in Figure 3.

Arriving trains unload their coal to the terminal via one of two inloaders. An inloader is a facility for transferring coal to the conveyor network of the terminal. The inloaders work in parallel and share an access rail where trains queue until one of the inloaders becomes available. After unloading, each train will be sent to the train depot before starting its next roundtrip.

Each company has a set of designated stockpiles in the terminal stockyard. There is also a Cargo Assembly Area (CAA): a set of stockpiles that could be temporarily allocated to a user or product. The boundaries of stockpiles might be considered fixed for short term planning but they could change over time. There are three types of yard machines: stackers, reclaimers and dual-function machines. Stackers transfer coal from conveyor belt system to stockpiles. Reclaimers unload stockpiles and transfer coal back to the conveyor belt system. Dual-function machines can carry out both tasks and most yard machines are dual-function machines.

There are many operational details of how a stacker-reclaimer might work including stacking mode (cone or window), reclaimer mode (pilgrim or bench), the position of the boom, safety distances relative to other stackers/reclaimers, and scheduled/unscheduled maintenance. However, in this research two main constraints are specially considered: two stacker-reclaimer machines cannot work on a same stockpile simultaneously, and machines working on a same bund (i.e. single track between rows) have a minimum separation and cannot pass each other. We refer to these constraints as stacker-reclaimer conflicts.

Unloading of each train is a concurrent activity by various resources. Inloaders transfer mining materials to a network of conveyor belts, which takes the coal near to an appropriate stockpile. A stacker then stacks the coal onto a specified stockpile. The conveyor network has a high level of capability. It can carry large volumes of coal to various stockpiles so that in practice it is seldom a constraint. It can support two stacking operations at two stockpiles and four reclaiming activities at four stockpiles simultaneously.

A ship that is waiting in the port vicinity is arranged to berth once all required loads (or parcels)
have become ready for outloading from stockpiles. There are, however, exceptions in that some stocks might become ready at stockpiles after berthing but just prior of their transport to the ship. There is also the possibility of directly transferring stock from a train to a ship without being stored on a stockpile. This is called direct loading. After a ship has berthed, coal is reclaimed from an appropriate stockpile by a reclaimer to the conveyor network and then is transferred to ship. To increase ship loading speed, typically two reclaimers are used to feed allocated conveyors simultaneously but this may be reduced to one under various circumstances.

The outloading process includes various practical considerations. One is the existence of a surge bin on the outloading network which stabilises the fluctuations in the amount of arriving products. The loading operations generally follow to a loading plan and should be coordinated with the ship’s master. The ship’s master controls how the loading procedure should take place (loading plan). The loading procedure is very important since an incorrect procedure would not only unbalance the ship but also might exert forces that are quite harmful for the ship’s structure.

One particular constraint that has been considered in the loading of a ship is tidal constraints. Ships cannot berth or sail at low tide. If the completion of loading operation is delayed and a ship misses the deadline, it must wait until the arrival of the next high tide.

5.1. An Integer Programming Model for the Stockyard System

We have developed the following integer programming (IP) model for the allocation of product to stockpiles. Our model is based on variables that are related to events over time. Time is discretised and so variables are defined over discrete sets. The duration of a time unit reflects the desired scheduling accuracy (we project that an hourly basis provides enough insights for practical purposes). We will use the notation \([t, t']\) to represent time intervals, where instant \(t\) is included but instant \(t' > t\) is excluded from the interval.

The basic inputs are the type of the product, and the opening and closing times for stockpiles, stackers and reclaimers. The stockpiles and their locations are identified with \((r, m)\), i.e. row and meterage. There are three types of opening and closing time intervals. These types are distinguished by superscript and are indexed by \(i\) in the following:

- \([to^1_i, tc^1_i]\) — an interval beginning at the time a stockpile becomes ready to receive coal for storage, and ending at the time that this stockpile is available to receive a new product.
- \([to^2_i, tc^2_i]\) — an interval over which a stacking operation occurs.
- \([to^3_i, tc^3_i]\) — an interval over which a reclaiming operation occurs.

For example, consider we have a list of products \(P = \{1, 2, 3, 2\}\). There will be a corresponding list of stockpile opening-closing times \([to^1_i, tc^1_i]\) given by \(T_{OC}^1 = \{(1, 15), (3, 18), (9, 21), (12, 21)\}\). In this terminal, different products have pre-designated stockpile locations, partly because the terminal is owned by several different companies. So, for example, a particular product may be permitted in stockpiles at locations \((r, m)\) \(\in\ \{(1, 10), (1, 12), (2, 7), (2, 4)\}\). We define \(S\) to be the union (for all of the elements of \(P\)) of these sets of permissible stockpile locations.

- \(S_{P_i}\) : the set of possible stockpiles for a product \(P_i\).
• \( \mathcal{K}_{r,m} \): the set of machines that can access \((r, m)\) stockpile.

• \( \mathcal{K}_j \): the set of machines allocated to row (bund) \(j\).

• \( d_{k,m} \): the distance of a stockpile from the end of its row, in the direction of machine \(k\) (there are generally two stacker/reclaimer machines in each row).

We define zero-one decision variable \( Y_{r,m,i,t} \) to be one, if the \(i\)-th item in product list is allocated to position \((r, m)\) in the stockyard at time \(t\). Also \( X_{k,m,i,t} \) is a zero-one variable to indicate whether \(i\)-th opening-closing for stacking machines is allocated or not. \( Z_{k,i,t} \) is a similar variable for the opening and closing of reclaiming. \( C \) variables, on the other hand, are related to conflicts. The formulation below is designed so that physical infeasibilities can scheduled but are minimised.

\[
\text{Min } \sum (C_1^t + C_2^t + C_3^t) 
\]

\[
\sum_r \sum_m Y_{r,m,i,to^1_i} = 1 \quad \forall i \text{ and } (r, m) \in S_P \quad (2)
\]

\[
Y_{r,m,i,t} = Y_{r,m,i,to^1_i} \quad \forall r, m, i \text{ and } t \in [to^1_i, tc^1_i) \quad (3)
\]

\[
\sum_k \sum_m X_{k,m,i,t} = Y_{r,m,i,to^1_i} \quad \forall i, t \in [to^2_i, tc^2_i) \text{ and } k \in \mathcal{K}_{r,m} \quad (4)
\]

\[
\sum_k \sum_m Z_{k,m,i,t} = Y_{r,m,i,to^1_i} \quad \forall i, t \in [to^3_i, tc^3_i) \text{ and } k \in \mathcal{K}_{r,m} \quad (5)
\]

\[
\sum_i Y_{r,m,i,t} \leq 1 + C^1_t \quad \forall r, m, t \quad (6)
\]

\[
\sum_m \sum_i (X_{k,m,i,t} + Z_{k,m,i,t}) \leq 1 + C^2_t \quad \forall k, t \quad (7)
\]

\[
\sum_{k \in \mathcal{K}_j} \sum_m \sum_i d_{k,m}(X_{k,m,i,t} + Z_{k,m,i,t}) + MS \leq BL + C^3_{t,j} \quad \forall t \text{ and } j = 1, \ldots, |r| \quad (8)
\]

Equations 2 and 3, allocate a product (parcel) to a location \((r, m)\) and retain the allocation throughout the opening and closing of that parcel. Equation 4 allocates a stacking operation to a machine, while Equation 5 allocates a reclaiming operation to a machine. The next three constraints deal with the number of possible conflicts. Equation 6 measures the possible number of conflicts in assigning different overlapping parcels to a single stockpile. Equation 7 calculates the number of conflicts in assigning overlapping reclaiming and stacking tasks to a single machine. The final constraint determines the amount of conflict in machines-on-the-same-bund movement (where \(MS\) is the minimum separation between machines and \(BL\) is the bund length). By introducing different weight coefficients, it is possible to change the relative importance of different types of conflicts.

Although we have managed to formulate this model for the stockyard system, the solution procedure seems to be impractical for realistic applications. We applied a version of the model to small size problems. The solution time appears to increase very rapidly with the size of problems. However, improvements to the formulation and the use of techniques such as cutting planes might yield methods which are practical.
5.2. A Greedy Heuristic for Stockyard System

The above model is relatively complicated and difficult to solve quickly. The use of a general integer programming solver generally results in long computation times which are potentially quite variable depending on the data in the instance. Hence we use a heuristic approach to solve this problem. A feasible solution to the IP can be constructed relatively simply using a greedy algorithm. The method we use is as follows:

1. Sort the parcels in chronological order (see below).
2. For each parcel in the list determine the possible set of locations where space for the parcel can be allocated based on product restrictions.
3. Eliminate choices that are no longer possible because other parcels have already been allocated in the stockyard.
4. If there is at least one location that would not lead to stacking or reclaiming conflict with parcels already allocated (based on expected railing times) then eliminate all locations leading to conflict.
5. If there are multiple locations remaining choose one that had the same product stored in that location previously or has the same product allocated to adjacent locations.

Assigning parcel locations in the order in which they are to be shipped (in Step 1) makes good sense. Not having stockyard areas allocated so as to allow parcel stacking and reclaiming without conflict, for parcels to be shipped early in the planning horizon, is almost certainly going to lead to delays in shipping and hence reduced supply chain efficiency.

This constructive heuristic allows the very rapid construction of a solution which is not necessarily optimal. The heuristic could be repeated iteratively, with list randomisation, in a search for improved solutions. In a more sophisticated approach the heuristic could, for example, be embedded into a GRASP (Greedy Randomized Adaptive Search Procedure) or a problem-space search procedure. However, the heuristic performs adequately in its intended industrial context, and so we have not experimented with more elaborate algorithms.

6. SIMULATION APPLICATION

In order to test the above models we developed a simulation that runs for a full year, taking into account a variety of factors found in the real application such as breakdowns, machine maintenance, delays and the like. We found that the mixed integer programming approach resulted in modules that were too cumbersome and slow for the purpose of simulating a whole year (where each of the models would need to be solved many times over). Hence we used an alternative approach that combined simulation with heuristic solution techniques.

We have attempted to replicate the major features of the entire coal export supply chain system in the simulation. As well as the railing system and stockyard system, the shipping system is a critical part of the supply chain. The shipping system involves management of the ship ‘arrival stream’ and the procedures for ship berthing and sailing.
The times of berthing and sailing, the length of loading times, also the heights of ships’ drafts are particularly critical because of tidal restrictions. The terminal under study experiences substantial tides. At low tide it is frequently unsafe for ships to approach or move (particularly when departing from the terminal). Tide level varies with time, which makes schedule execution vulnerable to disruptions caused by ships missing their planned tide.

The loading of a ship is a complex procedure and it is affected by decisions made within the terminal as well as by ship masters. The loading should carefully maintain the ship’s balance so as to avoid damage to the structure of ships. The modelling of such complex procedures in terms of mixed integer programming is very difficult. By contrast, simulation provides an opportunity to capture most of the details that are important in the real system, and the model can be iteratively refined and enhanced with optimisation. Refinement in this sense is continuous effort towards capturing realism, and optimisation is the integration of decision modules which incorporate optimisation algorithms and undertake (localised) decision making. In the remainder of this section we explore a partial application of simulation for the system and show that such an approach is useful.

A major overall objective for the supply chain is to load ships as quickly as possible and to avoid delays. This objective is summarised in demurrage. Demurrage is a penalty payment associated with a ship loading time that exceeds an agreed duration called the lay time. This penalty compensates ships for unproductive time. To avoid demurrage, planned quantities of products should be available in the stockyard prior to loading opportunities on waiting ships. The opposite measure is despatch, which is a bonus paid when a ship is served earlier than expected.

Two major supply chain issues are the management of coal inventory and the deployment of the required number of trains. However, the terminal has limited capacity in terms of space, inloading rate (transportation of coal to stockpiles) and outloading (transportation and loading of coal from stockpiles to ships). Moreover, the terminal is a user-owned facility and is required to provide a fair share of time and equipment resource to each owner, on a monthly basis referred to as user entitlements.

![Figure 4. A comparison of costs associated with three railing policies over a year](image)

As discussed in Section 4.1., three policies (heuristic rules) have been considered for the railing decision module: railing to entitlement, railing to shipping, and campaign railing. In addition,
the stockyard layout is re-optimised at regular intervals by a decision module that uses the greedy heuristic. We examined the three policies for a simulated time of a year (365 days, 24 hours per day), using data sourced from historical information about the coal terminal’s operations. The data encompassed all terminal equipment and their capacities, the coal demands, ship arrival operations and other relevant information including weather, breakdowns, and maintenance requirements.

There are many measures that one could use to compare possible railing policies such as average queuing time for ships, maximum number of queue length, and so on. However, we only consider the despatch incentive and demurrage cost. Figure 4 compares demurrage and despatch for the three different railing policies using scaled costs (real figures on costs cannot be shown because of their commercial implications).

Railing to entitlement provides high despatch benefits but incurs very high demurrage costs as well. In this case, demurrage almost cancels out the despatch benefits. Railing to shipping, on the other hand, provides slightly more benefits in the form of despatch but it results in much less demurrage costs. The demurrage cost is almost sixty percent of total despatch benefit. For campaign railing, the amount of despatch increases even more while demurrage cost is even less. In this case, demurrage costs is only thirty percent of total despatch benefit. This clearly shows that if users (mining companies) could share their entitlements and move toward policies that are more flexible, they would help the terminal to better achieve its delivery goals. This in turn implies that the terminal could provide a higher throughput.

7. CONCLUSION

This paper has highlighted some of the industrial requirements, research opportunities and modelling issues which are relevant to coal export supply chains. We have examined interdependent subsystems of a coal export supply chain and we have discussed the integration of the subsystem models into a discrete event simulation system. As part of our examination we have developed integer programming models for the railing system and for stockyard management. These models have not previously been introduced in the literature and they can be successfully applied to solve small instances of their respective problems. The integration of these models in an industrial-scale simulation or optimisation context, however, does not appear computationally viable. As a consequence we have also developed heuristic methods that execute quickly and which can be embedded in randomised search techniques when practical execution time-limits permit. We have embedded these heuristics into a detailed and expansive simulation of the coal export supply chain. Our ongoing work indicates that the embedding of customised heuristics into simulations provides for detailed studies of subsystem integration protocols and practices, can demonstrate the potential benefits of improvements in decision-making systems (in both the simulated and real environments), and can substantially inform real-world investment decisions.

REFERENCES


